NASA Technical Paper 3639

September 1996

A Formal Algorithm for Routing Traces on a Printed Circuit Board

David R. Hedgley, Jr.

NASA Technical Paper 3639

1996

A Formal Algorithm for Routing Traces on a Printed Circuit Board

David R. Hedgley, Jr.
Dryden Flight Research Center
Edwards, California

National Aeronautics and Space Administration

Office of Management

Scientific and Technical Information Program

ABSTRACT

This paper addresses the classical problem of printed circuit board routing: that is, the problem of automatic routing by a computer other than by brute force that causes the execution time to grow exponentially as a function of the complexity. Most of the present solutions are either inexpensive but not efficient and fast, or efficient and fast but very costly. Many solutions are proprietary, so not much is written or known about the actual algorithms upon which these solutions are based. This paper presents a formal algorithm for routing traces on a printed circuit board. The solution presented is very fast and efficient and for the first time speaks to the question eloquently by way of symbolic statements.

NOMENCLATURE

The following nomenclature do not appear in alphabetical order as each definition is predicated upon previous definitions.

μ the major axis (horizontal or vertical direction) where the sequence $\{J_i\}$ is associated with μ.

β the minor axis (horizontal or vertical direction) where the sequence $\{M_{k_i}\}$ is associated with β.

RM a rectangular matrix of evenly spaced grid points where a node is defined to be an arbitrary grid point belonging to RM.

RM_h RM represented as a sequence of horizontal lines composed of grid points.

RM_v RM represented as a sequence of vertical lines composed of grid points (fig. 1).

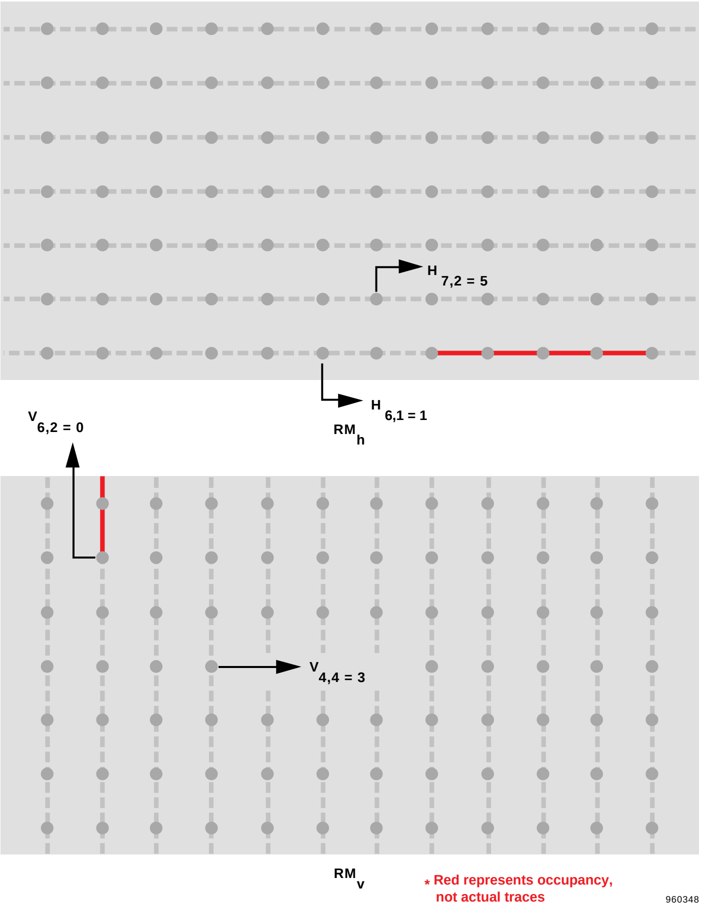

Figure 1. A representation of an internal matrix.

$H_{p,l}$ the number of available nodes from left to right at the node, (p, l), that belongs to RM_h (fig. 1).

$V_{p,l}$ the number of available nodes from bottom to top at the node, (p, l), that belongs to RM_v.

J_i the i^{th} designated node on the major axis where sequence $\{J_i\}$ is monotonic and bounded. That is, if (IB, MB) and (IE, ME) are the initial and terminal nodes respectively, then $IB = J_1 > J_2 > J_3 > J_4 .....J_n = IE$, or $IB = J_1 < J_2 < J_3 < J_4 .....J_n = IE$. (These inequalities are consistent with the horizontal axis as the major axis. Otherwise, the sequence $\{J_i\}$ is bounded by MB and ME, and the inequalities will be consistent with the vertical axis as the major axis. Moreover, $|J_i - J_{i-1}|$ represents the number of available nodes between J_i and J_{i-1}. That is, the sequence $\{J_i\}$ is determined by availability (fig. 2). If IB = IE or MB = ME, then only one major axis (direction) exists.)

M_{k_i} the k^{th} designated node on the minor axis that intersects the major axis at J_i, where M_{k_i} may not be monotonic or bounded for any i. The sequence $\{M_{k_i}\}$ is designated by the assignment of nodes in a cluster, if possible (fig. 2).

$N_i \geq 1$ the number of nodes $(1 \leq k_i \leq N_i)$ on the minor axis associated with J_i. The M_{1_i} denotes the intersection node of the minor axis with the major axis at J_i.

$S(J_i, M_{k_i})$ the discrete function that first interrogates the availability of a path belonging to either RM_h or RM_v, which is initiated at the node (J_i, M_{k_i}), in the major direction. If applicable, the function then subsequently interrogates the availability of a path in the minor direction if, and only if, $J_{i+1} = IE$ or ME. The values of $S(J_i, M_{k_i})$ are defined as follows:

- NA = No availability (not any space on the major axis is free).
- PA = Partial availability on the major axis.
- TA = Total availability (both axes are free to complete the trace).

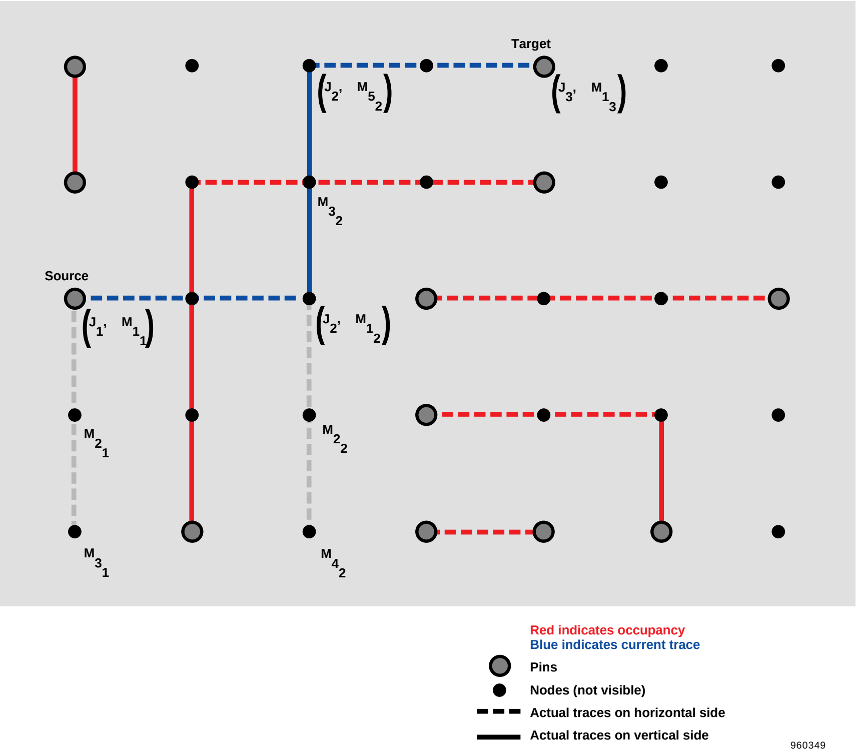

Figure 2. A representation of a typical trace.

NI no further interrogation, implying total success or total failure.

Definitions of Operations:

$C \leftrightarrows D$ the exchange of definitions of C and D.

$C \not\leftrightarrows D$ the exchange of definitions of C and D cannot take place.

$E[S(b, c)]$ evaluate $S(b, c)$.

INTRODUCTION

An automated solution to the printed circuit board problem that is both efficient and cost effective continues to be in great demand. Most of the present solutions[1–4] are either inexpensive but not efficient and fast, or efficient and fast but very costly. In either case, no solution exists that describes its algorithm symbolically with any degree of eloquence, nor are any algorithms sufficiently fast enough to facilitate parts placement for optimization. Many solutions are proprietary, so not much is written or known about the actual algorithms upon which these solutions are based.

The purpose of this paper is to accomplish three objectives:

- To facilitate optimization of arrangement of parts (for example, resistors and capacitors) on printed circuit boards for more sophisticated and expensive systems.
- To make a very fast and efficient algorithm accessible at no cost for individual users or small companies.
- To serve as a pedagogical tool that has heuristic value by presenting a formal symbolic structure that describes the entire algorithm.

FORMAL SYMBOLIC STATEMENTS

The purpose of the following algorithm, presented as a list of symbolic statements, is to construct a path from the source node, (IB, MB), to the target node, (IE, ME). A typical path (fig. 2) is described by the sequence of nodes (J_1, M_{1_1}), (J_2, M_{1_2}), (J_2, M_{5_2}), and (J_3, M_{1_3}). In this case, $M_{1_1} = M_{1_2}$ and $M_{5_2} = M_{1_3}$. In general, if (J_i, M_{k_i}) is an arbitrary element in a sequence leading to a solution, then the next element in the sequence will be either $(J_{i+1}, M_{1_{(i+1)}})$ or $(J_i, M_{(k+u)_i})$, where $u \geq 0$. Thus, the algorithm is as follows:

(a) $N_i > 1, k_i < N_i,$ and $S(J_i, M_{k_i}) = NA \Rightarrow E[S(J_i, M_{(k+1)_i})]$

(b) $i > 1, N_i \geq 1, k_i = N_i,$ and $S(J_i, M_{k_i}) = NA \Rightarrow E[S(J_{i-1}, M_{(k+1)_{i-1}})]$

(c) $i = 1, N_i \geq 1, k_i = N_i,$ and $S(J_i, M_{k_i}) = NA \Rightarrow \mu \leftrightarrows \beta$ or $(IB, MB) \leftrightarrows (IE, ME)$ and $E[S(J_1, M_{1_1})]$

(d) $i = 1, N_i \geq 1, k_i = N_i, S(J_i, M_{k_i}) = NA, \mu \not\leftrightarrows \beta$ and $(IB, MB) \not\leftrightarrows (IE, ME) \Rightarrow NI$

(e) $S(J_i, M_{k_i}) = PA \Rightarrow E[S(J_{i+1}, M_{1_{(i+1)}})]$

(f) $S(J_i, M_{k_i}) = TA \Rightarrow NI$

The efficiency of the algorithm is also predicated upon the process that determines the availability of rows and columns and determines the maintenance of the two respective directories. That is, a directory of rows, RM_h, and a directory of columns, RM_v, are maintained where each directory theoretically addresses different sides of the routing board. If required, any subsequent layers or surface mounts are handled similarly.

In general, the determination of the availability at an arbitrary node, (p, l), with respect to either RM_h or RM_v may be addressed accurately, keeping in mind that $H_{p,\,l}$ and $V_{p,\,l}$ are interchangeable as they relate to the following explanation (fig. 1). For the sake of specificity, $H_{p,\,l}$ will be employed here. Thus, let (u_o, l_o) be a target with respect to (p_o, l_o) and where $u_o > p_o$. Then the availability at $(p_o, l_o) = \min[H_{p_o,\,l_o}, u_o - p_o]$. However, if $u_o < p_o$, then construct the equation $T_o = H_{u_o,\,l_o} - (p_o - u_o)$. Consider the two cases $T_o \geq 0$ and $T_o < 0$. If $T_o \geq 0$, then the availability at (p_o, l_o) with respect to (u_o, l_o) is $p_o - u_o$. If $T_o < 0$, let $T = H_{u,\,l_o} - (p_o - u)$, where $u_o < u < p_o$. Then the availability at the node (p_o, l_o) that belongs to RM_h is calculated using the bisection method whose direction of seek is governed by the sign of T. Moreover, because the maximum availability is sought, the number of iterations will be $n = [LOG_2(p_o - u_o)] + 1$.

The integrity of the algorithm, which is implemented by recursive descent, is not disturbed by either the definition of the major and minor axes or the specificity of the terminal and initial nodes. Therefore, all four combinations of axes and nodes are transparent to the algorithm itself. Furthermore, an additional criterion to interchange axes or nodes could be predicated on the value of i or the magnitude of k_i for some i.

The number of vias can be minimized from the contents of the directories $(RM_v$ and $RM_h)$ after the solution is complete. This reduction results in a lower cost for the actual physical implementation of the solution. Finally, the updating of the directories dynamically is trivial, as the RM_v and RM_h contents are corrected based on the values of J_i and the corresponding M_{k_i} resulting from the monotonic sequence $\{J_i\}$.

The application of the theory presented confirms the objectives sought. Appendix A presents examples of typical solutions found. Moreover, a comparison study of the state-of-the-art algorithms is made. Appendix B addresses the efficiency, the cost, the ratio of solution to the Manhattan distance, and the central processing unit time.

CONCLUSION

An automated solution to the printed circuit board problem that circumvents the devastation of exponential growth as a function of complexity has been presented. Moreover, application of the theory displayed in Appendix A demonstrates its versatility and range. The statistics presented in Appendix B confirm the objectives sought. The computer program based upon the ideas presented here will be offered by COSMIC, located in Athens, Georgia.

The author wishes to acknowledge Glenn Bever, Rodney Bogue, and Harry Chiles for their many helpful suggestions.

APPENDIX A
EXAMPLES OF TYPICAL SOLUTIONS[*]

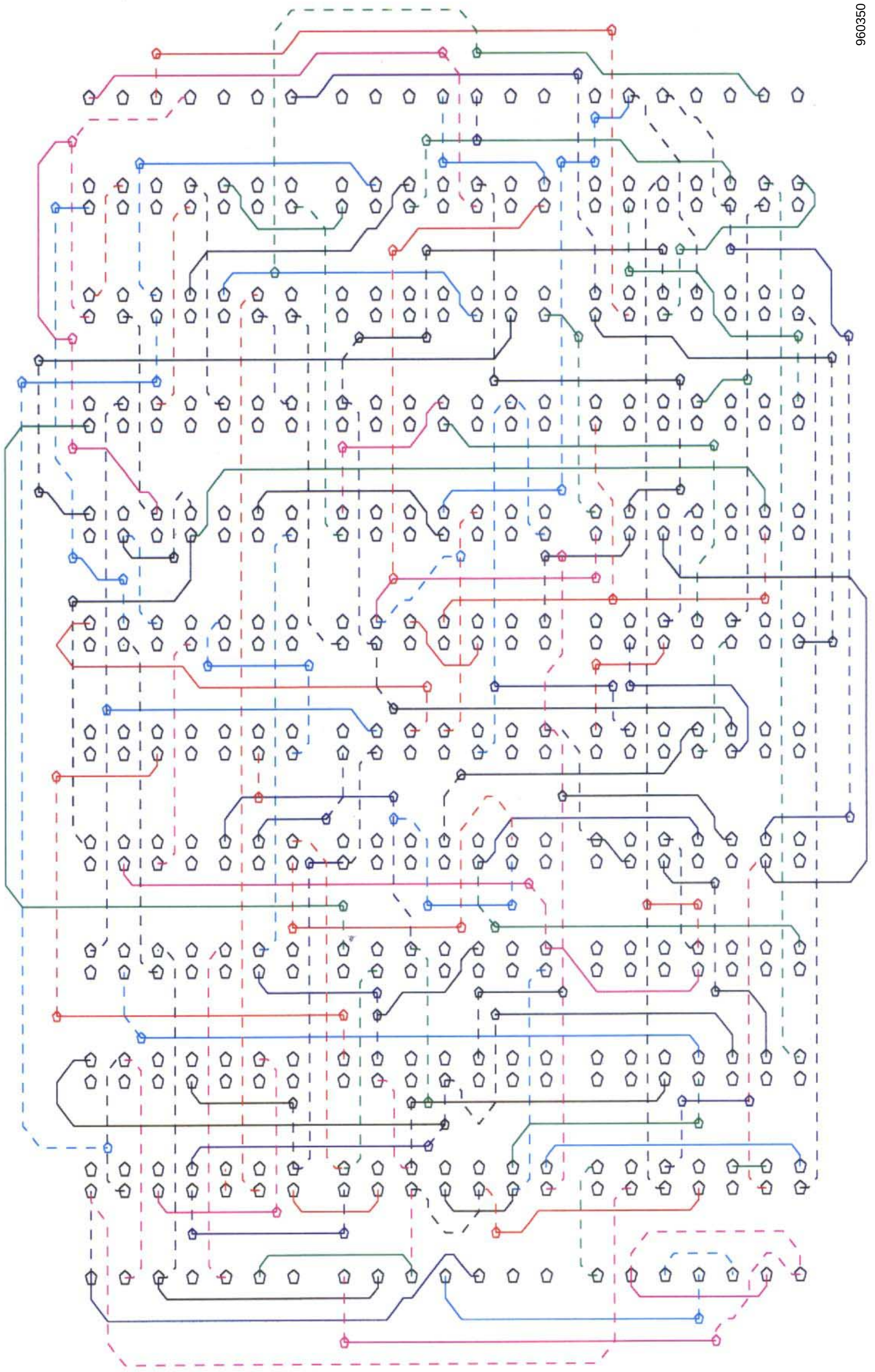

Figure A-1. Board without grid interference considerations; grid size = 0.05 in.

[*] Dash lines and solid lines are on different sides of the layer.

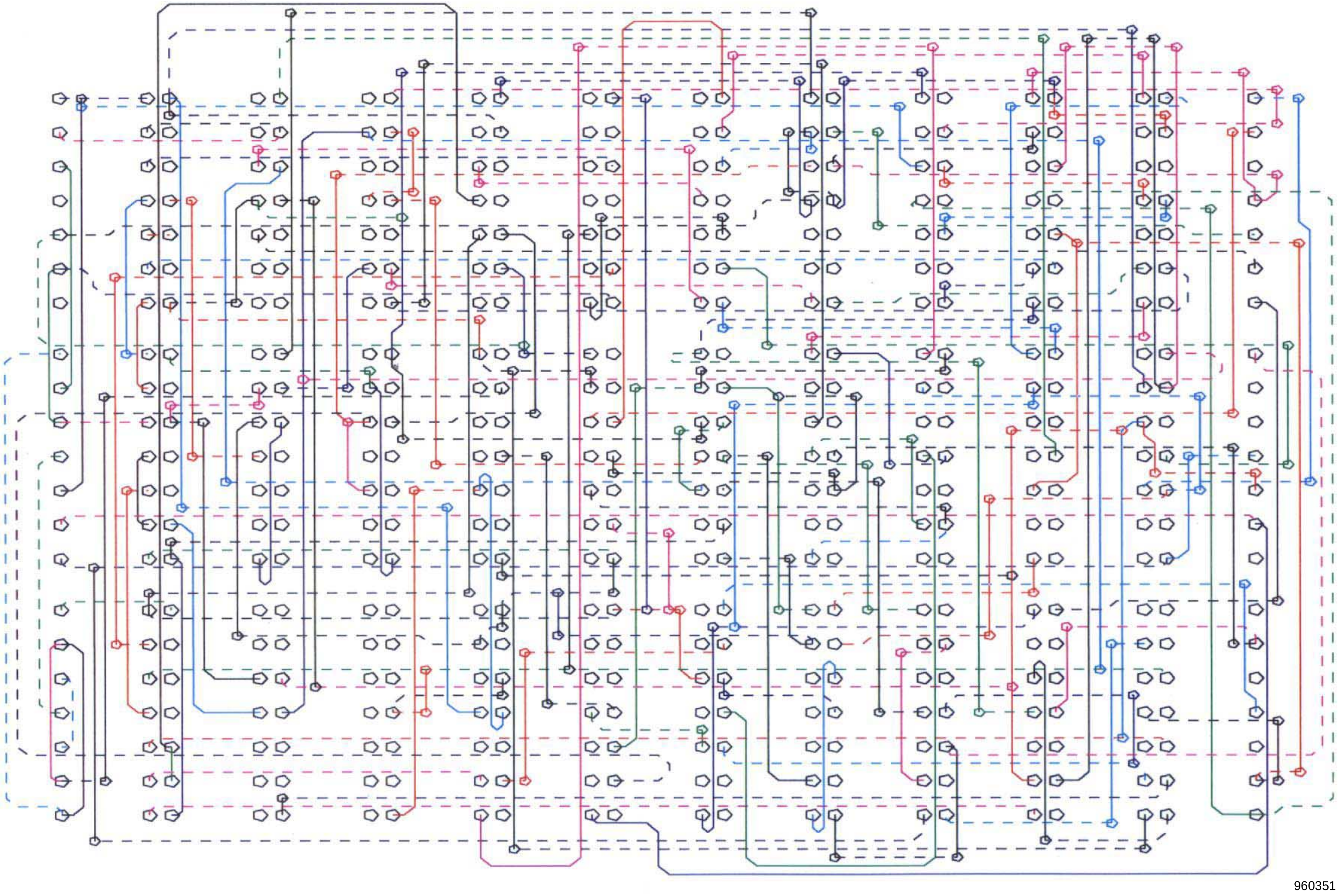

Figure A-2. Board with grid interference considerations; grid size = 0.025 in.

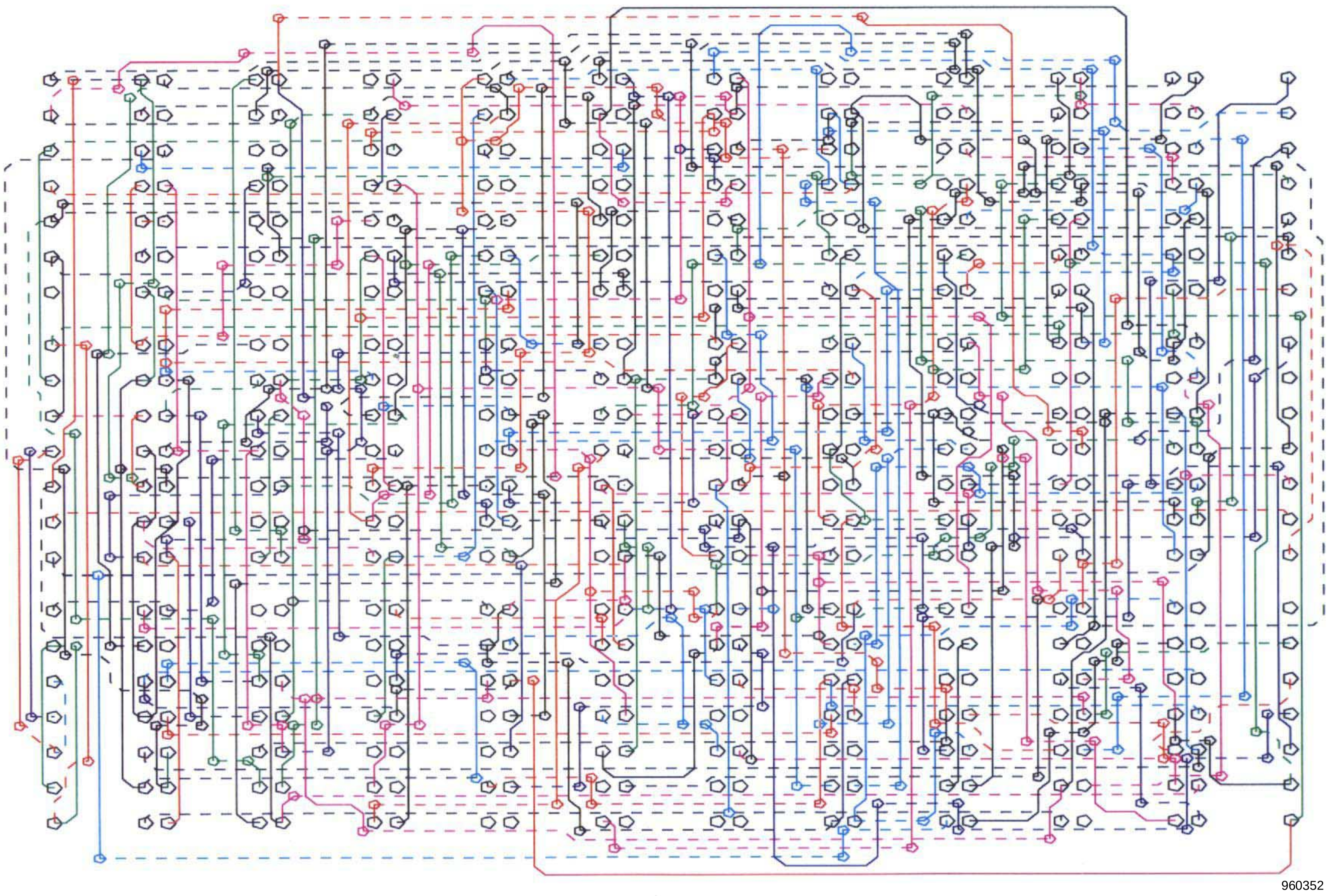

Figure A-3. Board without grid interference considerations; grid size = 0.025 in.

APPENDIX B
TABLE OF COMPARISONS*

Algorithm	Ratio, actual distance/ Manhattan distance	Completion rate, percent	Cost, dollars	CPU time, minutes
Shape tech	1.05	98–100	12,000	10
Pcad	1.10	92	4,000	30
Hedgley	0.97	93	0	0.25

*Chart statistics are reflective of a one-layer board that is two-sided with equivalent grid sizes.

REFERENCES

[1] Kinnucan, Paul, “Printed-Circuit Board Design: Low-Cost Systems are Winning Respect,” *The S. Klein Computer Graphics Review*, Summer 1987, pp. 79–105.

[2] Wittenberg, Roland C., “Algorithms Do What Designer Does Naturally,” *Electronic Engineering Times*, September 14, 1987, pp. 137–138.

[3] Jackoway, Gary, “Automating the Printed Circuit Board Design Process,” *Hewlett-Packard Journal*, February 1988, pp. 68–71.

[4] “Speedy Gridless Software Designs Denser PC Boards,” *Electronics*, v. 60, no. 14, July 9, 1987, pp. 102–103.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
September 1996

3. REPORT TYPE AND DATES COVERED
Technical Paper

4. TITLE AND SUBTITLE
A Formal Algorithm for Routing Traces on a Printed Circuit Board

5. FUNDING NUMBERS
WU 505-68-84

6. AUTHOR(S)
David R. Hedgley, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Dryden Flight Research Center
P.O. Box 273
Edwards, California 93523-0273

8. PERFORMING ORGANIZATION REPORT NUMBER
H-2104

9. SPONSORING/MONOTORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TP-3639

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified—Unlimited
Subject Category 61

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This paper addresses the classical problem of printed circuit board routing: that is, the problem of automatic routing by a computer other than by brute force that causes the execution time to grow exponentially as a function of the complexity. Most of the present solutions are either inexpensive but not efficient and fast, or efficient and fast but very costly. Many solutions are proprietary, so not much is written or known about the actual algorithms upon which these solutions are based. This paper presents a formal algorithm for routing traces on a printed circuit board. The solution presented is very fast and efficient and for the first time speaks to the question eloquently by way of symbolic statements.

14. SUBJECT TERMS
Artificial intelligence, Printed circuit board, Routing

15. NUMBER OF PAGES
13

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
Unlimited

Available from the NASA Center for AeroSpace Information, 800 Elkridge Landing Road, Linthicum Heights, MD 21090; (301)621-0390